## REMARKS/ARGUMENTS

By this paper, Applicant responds to the Office Action of April 3, 2006 and respectfully requests reconsideration of the application. The shortened statutory period runs through June 3, 2006. Accordingly, this response is timely.

Much of this paper is a near-verbatim copy of the paper previously considered by an Appeal Conference in September 2005. The issues are nearly identical: the "Description of the Preferred Embodiments" section of the primary reference now at issue, Robinson '095 (6:56 to 90:12) is word-for-word identical to the Description of the reference previously considered by the Appeal Conference, Chernoff '028 (7:18 to 90:32), with the exception of a few isolated word substitutions and typographical changes. The Appeal Conference has already reviewed and reversed a number of the positions raised in the Office Action.

### I. Real Party in Interest

The real party in interest is ATI International SRL of Barbados, the assignee of this application. ATI International is related to ATI Technologies, Inc. of Ontario, Canada.

### II. Related Appeals and Interferences

Applicant is unaware of any related appeals or interferences.

### III. Status of Claims

Claims 1-83 are now pending, a total of 83 claims. Claims 1, 5, 33, 56 and 79 are independent.

### IV. Status of Amendments

No amendments have been proposed since the "final" Office Action of April 3, 2006.

### V. Summary of the Subject Matter

This § V presents a primer in the technology. The subject matter of specific claims is discussed with respect to each claim in § VI.

### A. Background: the Robinson '095 Patent

The Robinson '095 patent is a useful example of several relevant computer engineering concepts.

The Office Action points to three unrelated portions of the Robinson '095 patent. The first, 14:65 to 19:57, relates to a software interpreter for an X86 running on an Alpha (a RISC processor developed by Digital Equipment Corp.). The second, 25:10-60, discussing Fig. 21, relates to multi-threading of X86 tasks, that is, changing from one software thread to another, but staying in simulated X86 mode. The third, at 25:62 to 33:60, discussing Figs. 22-26, relates primarily to transitions between X86 simulated execution and Alpha native execution, and subroutine calls, neither of which involves a change of execution thread. The separateness of the sections is easily seen. For example, "contexts" and the "context data structure" 180 and multitasking, which are core to the discussion at 25:10-60, are never mentioned in Robinson's subroutine call section (25:62 to 33:60), and are described as being different things in Robinson's interpreter section (16:4-7).

#### 1. Background: Robinson '095, Multi-Tasking[1] and Context Switches

Since the early 1960's, computers have used "multi-tasking" to rapidly switch between two or more program processes, so that it appears that the computer is running several different programs simultaneously. For example, the computer on your desk allows you to run email, a word processor, and an internet browser simultaneously, and to switch between them rapidly. A user can switch from one program to another, while the first program continues to run.

A switch between programs, called a "context switch," is fairly analogous to switching between two paper-intensive tasks at your desk. When you switch tasks, you must carefully file away all of the papers of the old task and then retrieve the papers of the new task from their files, and lay them out on the desk for efficient access. You must make some record of what you were

---

[1] The concepts of "multi-tasking" and "multi-threading" are slightly different. Threads are less tied to particular hardware than are processes. Nonetheless, traditionally, it has been impossible to use a thread manager for one family of computers (Intel X86/Pentium, for example) to manage threads of a foreign computer (Digital Equipment Corp.'s Alpha, discussed in Robinson '095, or the Tapestry processor discussed in this application's specification). In the context of the issues now pending, the difference is immaterial, and "multi-threading" and "multi-tasking" can be considered as variants of each other.

doing when you set the old task aside, so you know where to resume. Similarly, when a computer switches programs, the full status of the old program must be carefully stored: this typically includes the values of the registers, any descriptors that describe how the processor accesses the memory of the program, and a pointer to the point at which execution is to resume. Then, all of the corresponding information describing the new task (which was previously saved, when that task was switched out) is loaded, so that control may be transferred to the new task.

Multi-tasking is typically implemented in a computer's operating system, typically at the lowest and most hardware-dependent layers. For example, Exhibit 1 (filed March 21, 2005) shows a very precise map for how an IBM mainframe computer stores every relevant piece of state in memory, and reloads new state from memory. Exhibit 2 (also filed March 21, 2005) shows the same kind of information for Intel's Pentium chip as of 1996. Digital's Alpha has a third form. The three are necessarily entirely different, because the three have entirely different hardware resources.

Therefore, traditionally, a scheduler is tied to one precise type of hardware. For example, even within the Intel chip family, operating systems are not interchangeable. One must match the operating system quite precisely to the processor being used: there was/is a separate version of MS-DOS and of Windows for each new "major" generation of Intel chip. A mismatch will result in a non-functioning system.

As will be discussed further in § V.A.3, below, all of the multi-tasking managers discussed in Robinson '095 are specialized to either Alpha or X86 – Robinson '095 takes care that his Alpha programs are only managed by his Alpha program manager, never the X86 program manager, and vice-versa.

### 2. Background: Robinson '095 and Subroutine Calls

At 25:62 to 33:60, discussing Figs. 22-26, Robinson '095 discusses mechanisms for subroutine calls and returns. The important aspect of this section is that the mechanisms discussed here are independent of the mechanisms discussed for multi-tasking and for the software interpreter of the X86. Good software systems maintain independence among different components, so that complex interactions are kept to a minimum. Robinson uses that basic design technique.

### 3. Background: Robinson '095 and Software Interpreters

Some manufacturers of computer hardware have found it valuable to allow their computers to run software that is coded in a foreign instruction set. For example, the Robinson '095 patent discusses a software system that allows Digital's Alpha processor to run Intel X86 application programs, using a "simulator" or an "interpreter." (Robinson '095, 6:66 to 7:2 and 14:65 to 19:56, discussing Figs. 7-12). An interpreter is software that creates a "virtual" CPU and non-native operating system.

Robinson maintains a strict wall of separation between his Alpha execution and his X86 execution – nothing will work right if, for example, Robinson tries to manage an Alpha program using his X86 program execution manager, or vice versa. Robinson's system provides a multi-threading capability for X86 programs separate from the multi-tasking and multi-threading capabilities that can be used to manage programs written for the Alpha (25:10-60).[2] Notably, all of the registers and structures, and "current state" mentioned in this section are X86 structures that are implemented as "virtual" software entities in the interpreter. There are no such registers in Robinson's Alpha hardware.

The context-switch capability of the Alpha operating system cannot context switch among Intel X86 threads, because Alpha uses different hardware structures than the X86 (*see* Robinson '095 2:1-5), and there is simply no way to access X86 registers in the Alpha instruction set, or vice-versa.[3]

---

[2] For example, Robinson '095 at col. 25:24-32 discusses saving and restoring the virtual X86 registers in a thread manager for X86 programs: "... the condition code bits [and] copies of the integer registers EAX 104a, EBX 104b, ECS 104c, EDX 104d EDI 104e, ESI 104f, EBP 104g and ESP 104h."

[3] It would be quite impossible to store or switch Alpha context using the mechanisms described in col. 25. For example, the Alpha processor has 31 integer registers and 31 floating-point registers that must be stored on context switch, each 64 bits wide (see the "Compaq Computer Corp., Compiler Writer's Guide for the Alpha 21264 (1999)," at pages 3-9 to 3-11, of record in this application, or at `//ftp.digital.com/pub/Digital/info/semiconductor/literature/cmpwrgd.pdf`). Alpha's data registers alone require almost 4000 bits of storage. In contrast, Robinson '095 teaches that his X86 "context data structure" only has space for the X86's eight 32-bit integer registers (col. 25:56-59; col. 16:32-33), a total of only 256 bits. 4000 bits of information cannot possibly fit in 256 bits of storage.

Robinson '095 discusses a number of other data structures that hold some context information. However, in every case, Robinson maintains a clear distinction – a given data structure holds X86 context and is managed only by software for managing X86 context, or holds Alpha context and is managed only by software for managing Alpha context.

### 4. Robinson's Three Major Components are Different

Notably, the interpreter's model of the X86 processor, the multitasking manager's model of processes, and the software for managing X86 <-> Alpha transitions are different things. For example, the "interpreter" of 14:65 to 19:56 models an X86 hardware microprocessor, while the multitasking manager of 25:10-60 manages software processes. The transition-on-subroutine call mechanism is different still, with its own data structures and code. While they cooperate with each other, they are each different things. Even though some components have similar names, they are still different things.

## B. The Embodiment in the Specification

With that background to establish some of the vocabulary, we can turn to the specification for this application. The specification discusses several techniques that allow the use of hardware and an operating system that are mismatched to each other.

Sections III and IV of the specification (pages 32-60) discuss mechanisms for allowing a non-native operating system, such as Microsoft Windows, to be used on hardware for which that operating system software has not been tailored, and to manage programs coded in the instruction set of that hardware. An introduction to this preferred embodiment appears at pp. 32-33 of the specification:[4]

> Referring to **Fig. 3a** and to Table 1, X86 threads (*e.g.*, **302**, **304**) managed by X86 operating system **306**, carry the normal X86 context, including the X86 registers... In addition, if an X86 thread **302**, **304** calls native Tapestry libraries **308**, X86 thread **302**, **304** may embody a good deal of extended context, the portion of the Tapestry processor context beyond the content of the X86 architecture. A thread's extended context may include the various Tapestry processor registers, general registers r1-r31 and r56-r63, and the high-order halves of r32-r55 (see Table 1), the current value of ISA bit **194**...
>
> The Tapestry system manages an entire virtual X86 **310**, with all of its processes and threads, *e.g.*, **302**, **304**, as a single Tapestry process **311**. Tapestry operating system **312** can use conventional techniques for saving and restoring processor context, including ISA bit **194** of PSW **190**, on context switches between Tapestry processes **311**, **314**. However, for threads **302**, **304** managed by

[4] This discussion, like all of § V, is a mere discussion of one concrete preferred embodiment, as a helpful aid to establish context for the discussion of the claims that follows. It is not a discussion of the claims themselves.

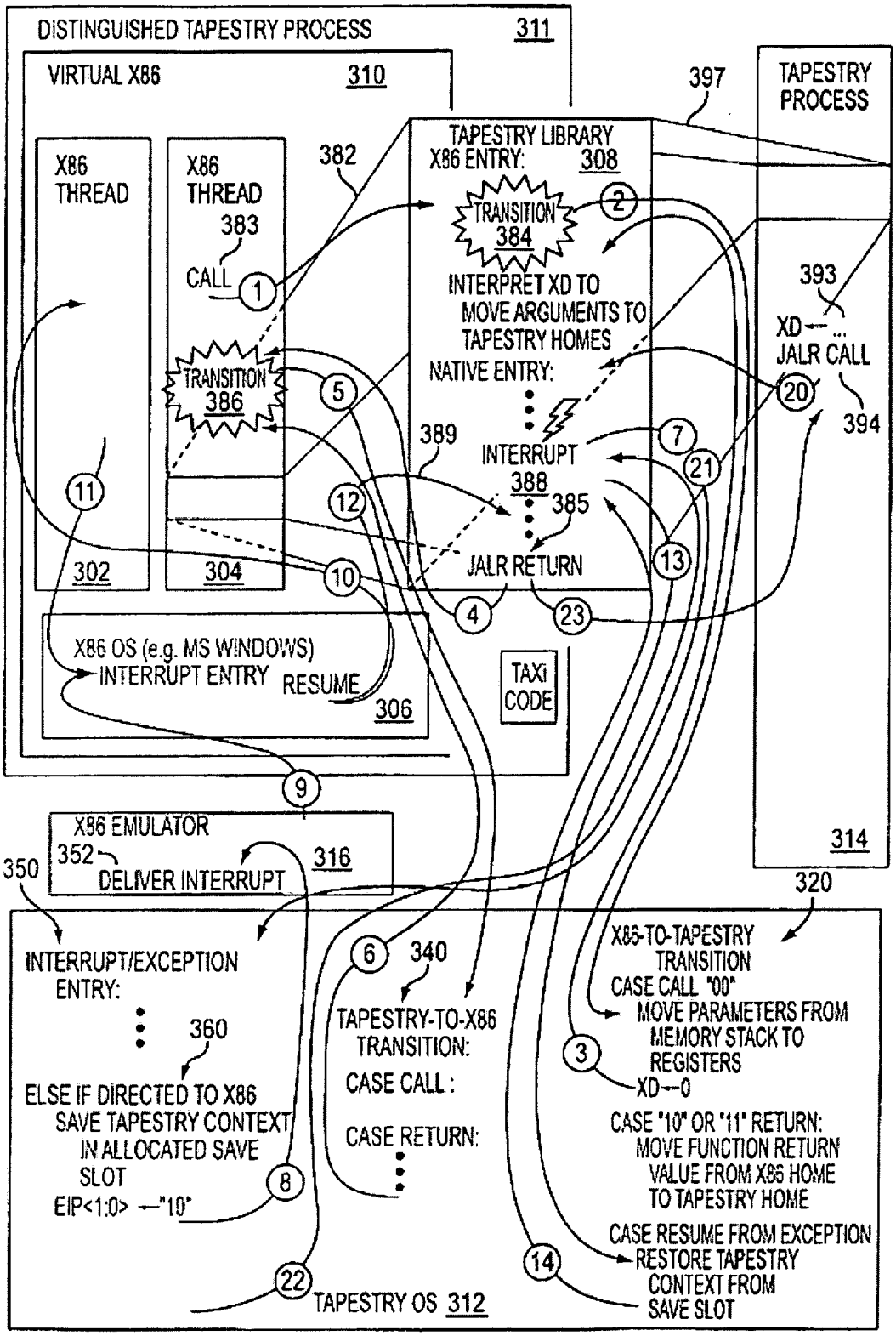


FIG. 3A

> an off-the-shelf X86 operating system **306** (such as Microsoft Windows or IBM OS/2) within virtual X86 process **311**, the Tapestry system performs some additional housekeeping on entry and exit to virtual X86 **310**, in order to save and restore the extended context, and to maintain the association between extended context information and threads **302**, **304** managed by X86 operating system **306**.
> ...
>
> **Figs. 3a-3n** describe the mechanism used to save and restore the full context of an X86 thread **304** (that is, a thread that is under management of X86 operating system **306**, and thus invisible to Tapestry operating system **312**) that is currently using Tapestry extended resources. In overview, this mechanism snapshots the full extended context into a memory location **355** that is architecturally invisible to virtual X86 **310**. A correspondence between the stored context memory location **355** and its X86 thread **304** is maintained by Tapestry operating system **312** and X86 emulator **316** in a manner that that does not require cooperation of X86 operating system **306**, so that the extended context will be restored when X86 operating system **306** resumes X86 thread **304**, even if X86 operating system **306** performs several context switches ... before the interrupted X86 thread **304** resumes. The X86 emulator **316** or Tapestry operating system **312** briefly gains control at each transition from X86 to Tapestry or back, including entries to and returns from X86 operating system **306**, to save the extended context and restore it at the appropriate time.

The specification of this application teaches mechanisms for allowing an operating system, such as Microsoft Windows™, to be used with hardware for which that operating system was not specifically designed. For example, the specification teaches how to use Microsoft Windows – which is available only on Intel chips and a very few other processors – on a foreign RISC (Reduced Instruction Set Computer) processor. This allows a totally new hardware design to use a mature and widely-available operating system, even if the source code is not available to allow Windows to be tailored to or customized to that RISC.

The particulars of the claimed subject matter are discussed in relation to specific claims.

## VI. Argument

### A. Standard of Review

As a matter of administrative law, agency decisions are void – that is, they have no legal existence – when an agency fails to make the showings required in the agency's own procedural handbook. *Service v. Dulles*, 354 U.S. 363, 388-89 (1957).

An agency action must be set aside if it is contrary to law, in excess of authority, short of statutory right, in excess of statutory jurisdiction, or without observance of procedure. 5 U.S.C. § 706(2). Review under these prongs of § 706 is non-deferential. *E.g.*, *Vitarelli v. Seaton*, 359 U.S. 535, 546-47 (1959) (Frankfurter, J., concurring) ("procedure must be scrupulously observed."); *Reuters v. F.C.C.*, 781 F.2d 946, 950-51 (D.C. Cir. 1986) ("*Ad hoc* departures from [an agency's] rules, even to achieve laudable aims, cannot be sanctioned"). Reviews on the merits are entirely non-deferential when agency or Administrative Procedure Act procedures have been violated at any stage of agency proceedings. *Stone v. Federal Deposit Insurance Corp.*, 179 F.3d 1368, 1376 (Fed. Cir. 1999) ("Our system is premised on the procedural fairness at each stage of [agency] proceedings. [A party before an agency] is entitled to a certain amount of due process rights at each stage and, when those rights are undermined, the [party] is entitled to relief."). The reason for non-deferential review in these circumstances follows from well-established principles of administrative law: an action taken in violation of an agency's own regulations is "illegal and of no effect," *Vitarelli*, 359 U.S. at 545; *IMS, P.C. v Alvarez*, 129 F.3d 618, 621 (D.C. Cir. 1997) (it is a "well-settled rule that an agency's failure to follow its own regulations is fatal to the deviant action"), and if no agency decision exists, then there is no agency action to which to give deference.

An agency decision must be set aside if it is "arbitrary [or] capricious" 5 U.S.C. § 706(2)(A). An agency action is arbitrary and capricious if it fails to consider relevant factors or makes an unexplained departure from past norms, *Motor Vehicle Mfrs. Ass'n, Inc. v. State Farm Mutual Auto. Ins. Co.*, 463 U.S. 19, 57 (1983) ("an agency changing its course must supply a reasoned analysis ..."); *Atchison Topeka & Santa Fe Rwy Co. v. Wichita Board of Trade*, 412 U.S. 800, 808 (1973). "Arbitrary and capricious" review is non-deferential when an agency has failed to address all relevant factors or all exceptions raised, has acted inconsistently with its own precedent, or has violated its own procedures. 5 U.S.C. § 557(c); *see Motor Vehicle Mfr's Assn.*, 463 U.S. at 48 ("an agency must cogently explain why it has exercised its discretion in a given manner"); *Atchison*, 412 U.S. at 806-07. Agencies only receive deference when their written decisions reflect a *bona fide* effort to engage in rational decision making.

The Board reviews an Examiner's factual and legal conclusions, and claim construction, without deference. *E.g.*, *Ex parte Toda*, 2001 WL 1729659 at *3 (BPAI Apr. 26, 2001). The initial burden to come forward with evidence always remains with the examiner, and the Board cannot affirm positions that the examiner has not articulated. *Ex parte Berg*, 2002 WL 32346092 at *2 (BPAI Feb. 6, 2003) ("the examiner must present a full and reasoned explanation of the rejection in the statement of the rejection, specifically identifying underlying facts and any supporting evidence, in order for appellants to have a meaningful opportunity to respond"); *Ex parte Schricker*, 56 USPQ2d 1723, 1725 (BPAI 2000). No burden shifts to the applicant until the Examiner has addressed every element of a *prima facie* case of unpatentability. *E.g.*, *In re Glaug*, 283 F.3d 1335, 1338, 62 USPQ2d 1151, 1152 (Fed. Cir. 2003) ("During patent examination the PTO bears the initial burden of presenting a *prima facie* case of unpatentability. If the PTO fails to meet this burden, then the applicant is entitled to the patent."); MPEP § 2142 (for obviousness, three *prima facie* showings must be made by the examiner before any burden shifts to an applicant).

### B. Paragraph 4 of the Office Action - § 112 ¶ 2 Issues

All claims were considered under § 112 ¶ 2 by the September 2005 Appeal Conference, and found to meet § 112 ¶ 2.

#### 1. Substantive Legal Standard for Definiteness

The legal standard for definiteness under § 112 ¶ 2 is met if there is any reasonable, unambiguous reading of a claim. *Exxon Research and Engineering Co. v. United States*, 265 F.3d 1371, 1375, 60 USPQ2d 1272, 1276 (Fed. Cir. 2001) ("If the meaning of the claim is discernible, even though ... reasonable persons will disagree, we have held the claim sufficiently clear to avoid invalidity on indefiniteness grounds.")

Slight deviations from perfect and pedantic "antecedent basis" is not a ground for rejection of claims. *Energizer Holdings, Inc. v. International Trade Commission*, 435 F.3d 1366, 1370-71, 77 USPQ2d 1625, 1628 (Fed. Cir. 2006) (it was reasonably ascertainable that "said zinc anode" referred to "anode gel" recited earlier in the claim – "When the meaning of the claim

would reasonably be understood ... the claim is not subject to invalidity upon departure from the protocol of 'antecedent basis.'").

### 2. Procedural Requirements for Raising an Indefiniteness Rejection

MPEP § 2173.02 states the minimum requirement for any rejection whatsoever to exist under § 112 ¶ 2 (emphasis added):

> **2173.02 Clarity and Precision [R-1]**
> The examiner's focus during examination of claims for compliance with the requirement for definiteness of 35 U.S.C. 112, second paragraph, is whether the claim meets the threshold requirements of clarity and precision, not whether more suitable language or modes of expression are available. ... he or she should allow claims which define the patentable subject matter with a reasonable degree of particularity and distinctness. Some latitude in the manner of expression and the aptness of terms should be permitted even though the claim language is not as precise as the examiner might desire. Examiners ... should not reject claims or insist on their own preferences if other modes of expression selected by applicants satisfy the statutory requirement.

Section 112, second paragraph, requires a "reasonableness" inquiry. For example, no rejection even exists where an artificially-constructed unreasonable or grammatically-incorrect reading of a claim is considered, while a reasonable and grammatically correct reading is ignored. MPEP § 2173.02 continues:

> [If] the language used by applicant satisfies the statutory requirements of 35 U.S.C. 112, second paragraph, but the examiner merely wants the applicant to improve the clarity or precision of the language used, the claim must not be rejected under 35 U.S.C. 112, second paragraph, rather, the examiner should suggest improved language to the applicant.

Again, the standard is simple: any reasonable interpretation of a claim is sufficient under § 112 ¶ 2. An examiner has no discretion to impose personal preference.

MPEP § 2173.05(e) makes clear that "antecedent basis" is not an automatic ground of rejection; there must be an additional showing, that the claim is unclear to one of ordinary skill:

> **2173.05(e) Lack of Antecedent Basis**
> Obviously, however, the failure to provide explicit antecedent basis for terms does not always render a claim indefinite. If the scope of a claim would be reasonably ascertainable by those skilled in the art, then the claim is not indefinite. ... Inherent components of elements recited have antecedent basis in the recitation of the components themselves. ... [T]he limitation "the outer surface of said sphere" would not require an antecedent recitation that the sphere has an outer surface.

### 3. The Claims Meet § 112 ¶ 2, and the Office Action is Insufficient to Raise Any Rejection

Paragraph 19(a) of the Office Action notes that "the operating system," "the thread scheduler," and "the operating system" unambiguously relate to earlier language in the respective claims. The Office Action makes no showing that the indicated claim language could reasonably be interpreted in more than one way. The Office Action answers itself, and no rejection is raised.

Paragraph 19(b) merely states that the terms "lack[ ] antecedent basis," but with no showing that the claim is unclear. No rejection exists.

| Office Action | Language Questioned | Antecedent Basis |
|---|---|---|
| ¶ 19(a)(i) | claim 1, line 2 "a pre-existing operating system" and claim 1, line 3, "the operating system" | Note 1 |
| ¶ 19(a)(ii) | | Note 1 |
| ¶ 19(a)(iii) | claim 5, line 2, "a pre-existing thread scheduler" and claim 5, line 4, "the thread scheduler" | Note 1 |
| ¶ 19(a)(iv) | claim 33, line 2, "a computer operating system" and claim 33, line 4, "the operating system" | Note 1, Note 2 |
| ¶ 19(b)(v) | claim 1, line 3, "the operating system" | claim 1, line 2 "a pre-existing operating system" |
| ¶ 19(b)(vi) | claim 5, line 4, "the thread scheduler | claim 5, line 2, "a pre-existing thread scheduler" |
| ¶ 19(b)(vii) | claim 43, line 3, "the context" | Note 3 |
| ¶ 19(b)(viii) | claim 46, lines 2-3 "the extended context" | claim 46, lines 3-4: "the extended context being the resources beyond those whose resource association with the process is maintained by the operating system" |
| ¶ 19(b)(ix) | claim 47, line 2, "the extended context" | |
| ¶ 19(b)(x) | claim 56, line 3, "the operating system" | claim 56, line 2 "a pre-existing operating system" |
| ¶ 19(b)(xi) | claim 79, line 3, "the service" | Note 4. |

Note 1. The Office Action itself notes that the claim is definite and unambiguous because there is only one reasonable antecedent in the claim. No rejection is stated.

Note 2. The Office Action deviates from the ordinary parsing of English sentences. This is not a proper basis for raising a rejection. No rejection is raised.

Note 3. Every process has a context, and thus "the context" has inherent antecedent basis in "the interrupted process." The claim is sufficiently definite. However, it could be improved

by changing "the context" to --the interrupted process' context-- and an examiner's amendment is authorized.

Note 4. "the service" is inherent in "a service routine" – a "service routine" must perform a "service" The claim is sufficiently definite. However, it could be improved by changing "of the service," to --of the service routine,-- and an examiner's amendment is authorized.

### C. Claims 33-36, 38-46, 50 and 52

Claim 33 is discussed in the context of Robinson '095 alone at ¶ 21 of the Office Action.

Claim 33 recites as follows:

> 33. A method, comprising:
>
> establishing an entry exception to be raised on entry to a computer operating system at a specified entry point or on a specified condition;
>
> **establishing a resumption exception to be raised on resumption from the operating system when such resumption is complementary to one of the specified entries;**
>
> on detecting a specified entry to the operating system from an interrupted process of the computer, raising and servicing the entry exception, and then entering the operating system to perform a service associated with the specified operating system entry; and
>
> **on detecting a complementary resumption, raising and servicing the resumption exception, and returning control to the interrupted process.**

Claim 33 is directed to wrapping a "specified entry point or on a specified condition" of a computer operating system (typically a conventional or preexisting entry point or service routine) in a pair of additional exceptions: one to be raised on entry, one to be raised on resumption. One example embodiment can be seen in Fig. 3a of this application, by following arrows ⑦, ⑧, ⑨, ⑫, ⑬ and ⑭. In this example, when an interrupt occurs (388 of Fig. 3a), normally execution would trap into the operating system (306 of Fig. 3a). Instead, a second exception (arrow ⑦ of Fig. 3a, corresponding to the "entry exception" of claim 33) is raised. This causes control to be transferred to a handler (350 of Fig. 3a, corresponding to "servicing the entry exception" of claim 33). When that handler returns control (arrow ⑨ of Fig. 3a), typically the operating system handles the original interrupt. When the operating system executes its return of control instruction (arrow ⑫ of Fig. 3a, corresponding to "a complementary resumption" of claim 33), another exception (arrow ⑬ of Fig. 3a, corresponding to the "resumption exception" of claim 33) is raised. This exception transfers control to a third handler (lower right corner of Fig. 3a, "servicing the resumption exception" of claim 33). The resumption exception handler returns

control to the original thread (arrow ⑭ of Fig. 3a, corresponding to the "returning control to the interrupted process" of claim 33). Thus, a typical embodiment of claim 33 would involve three exceptions and associated services or handlers. See Figs. 3a, 3h, 3i and 3j.[5]

### 1. First Error: Procedurally, the Office Action is Inadequate to Raise any Rejection

The first error in the Office Action is that it is incomplete. Because it only designates "portions relied on" and does not designate particular elements of the reference or otherwise "clearly explain the pertinence," as required by 37 C.F.R. § 1.104(c)(2), it is not clear how the Examiner proposes to match the reference to the claim, or even whether all limitations of the claim have been considered.

None of the portions of Robinson '095 (31:9-38, 26:29-55, 32:24-44, and 33:24-55) designated in the Office Action for the bolded portion of claim 33 use the term "exception" or "interrupt." The pertinence of the designated portions is not apparent, and the Office Action was required to provide a "clear explanation." What feature of Robinson '098 corresponds to the "resumption exception," and in what respect is it "complementary to one of the specified entries?" What in Robinson '095 corresponds to "raising and servicing the resumption exception" of claim 33? Have these claim limitations been considered, or have they been ignored?

Paragraph 14 of the Office Action then departs further from Rule 104, by asserting that "an exception or interrupt occurs," but without designating any portion of a reference, or otherwise identifying any support in the prior art. Rejections may not be based on rabbits pulled out of hats.[6] Unless some evidence can be identified, no rejection exists.

---

[5] This comparison of the specification to the claim is not a limiting description of the invention or the claim; it is merely offered as a concrete preferred embodiment to assist in understanding the claim.

[6] MPEP § 2143.03 restates Federal Circuit law: *every* limitation of a claim must be met by prior art drawn from one of the categories of § 102. No limitation may be rejected based on bald assertion. *In re Lee*, 277 F.3d 1338, 1343-44, 61 USPQ2d 1430, 1434 (Fed. Cir. 2002) ("'common knowledge and common sense' on which the Board relied in rejecting Lee's application are not the specialized knowledge and expertise contemplated by the Administrative Procedure Act. Conclusory statements such as those here provided do not fulfill the agency's obligation."); *Motorola v. Interdigital Technology Corp.*, 121 F.3d 1461, 1466-67, 43 USPQ2d 1490, 1490-91 (Fed. Cir. 1997) (every element must be met by prior art, even elements that are "well known" standing alone).

Because there has been no attempt to meet the requirements of 37 C.F.R. § 1.104(c)(2), no rejection exists.

### 2. Second Error: The Office Action's Analysis of Robinson '095 Relative to Claim 33 is Faulty

Nonetheless, in an effort to advance prosecution, Applicant observes as follows.

As the Office Action itself concedes, in Robinson '095 33:24-55, when a handler completes, it "executes a return instruction to return control to its caller routine." A smooth return using a "return instruction" is not an "exception."[7]

Robinson '095 31:9-38, 26:29-55, 32:24-44 and 33:24-55 do not mention an exception that could correspond to the "resumption exception." At most, any exceptions mentioned here might correspond to the "specified entry point or on a specified condition" in the first paragraph of claim 33.

Claim 33 distinguishes the art, and is therefore patentable.

### 3. Claims 34-36, 38-46, 50 and 52

Claims 34-36, 38-46, 50 and 52 are dependent on claim 33, and are therefore patentable with claim 33. In addition, the dependent claims recite additional features that further distinguish the art.

In the event that a future paper presents a complete statement of rejection, Applicant reserves the right to reply to the late examination, for example, designating claims that stand or fall separately.

---

If any rejection is raised in a future paper, Applicant requests a reference or affidavit in support of the "exception," pursuant to 37 C.F.R. § 1.104(d)(2), or that the portion of a reference relied upon be designated as nearly as practicable, as required by § 1.104(c)(2).

[7] Robinson '095 mentions exceptions in portions not designated in the Office Action. The Office Action makes no attempt to show that these exceptions are used in the manner recited in the claim, or relate to the designated portions of the references, and the correspondence or pertinence is not apparent.

### D. Claims 79-81

Claim 79 is discussed in the context of Robinson '028 alone at ¶ 29 of the Office Action. The portions of Robinson '098 cited now are essentially similar to the portions of Chernoff '028 previously considered by the Appeal Conference: they discuss the same figures and components.

Claim 79 recites as follows:

> 79. A method, comprising:
>
> during invocation of a service routine of a computer, passing a linkage return address to the service routine at which to resume execution on completion of the service, **the linkage return address being deliberately chosen so that an attempt to execute an instruction from the linkage return address on return from the service routine will raise a program execution exception**;
>
> on return from the service routine, **attempting to execute the instruction at the linkage return address** and raising the chosen exception; and
>
> after servicing the exception, returning control to a caller of the service routine.

One example is discussed in the specification at § IV.H, at page 49-50 of the specification.[8]

#### 1. First Error: The Office Action is Procedurally Incomplete – No Rejection Exists

The first error in the Office Action is the failure to "clearly explain the pertinence" of the designated portions of the references, as required by 37 C.F.R. § 1.104(c)(2), for example, by naming particular components of the reference as corresponding to the elements of the claim. In what respect is anything in Robinson '095 "deliberately chosen so that an attempt to execute ... will raise a program execution exception?" What feature of Robinson '095 corresponds to "attempting to execute the instruction at the linkage return address and raising the chosen exception?" Have these claim limitations been ignored?

The Office Action, ¶ 17, asserts that any one of the "native dynamic link", "native return address," or "non-native return address" satisfy the "linkage return address." However, the Office Action makes no attempt to show that any one of them satisfies all four appearances of

[8] This comparison of the specification to the claim is not a limiting description of the invention or the claim; it is merely offered as a concrete preferred embodiment to provide context for understanding of the claim.

the "linkage return address" in claim 79. Have these claim limitations and their interrelationships been ignored?

Without a clear and consistent statement of the Examiner's position, it is impossible to reply directly. For failure to comply with 37 C.F.R. § 1.104(c)(2), no rejection exists. Further, the arguments of § VI.D.2 were presented to the Examiner in December 2005. The Examiner failed to "answer all material traversed" as required by MPEP § 707.07(f). Any rejection that may have existed has lapsed for failure to observe Office procedure. Claim 79 is not rejected.

### 2. Second Error: The Office Action's Analysis of Robinson '095 Relative to Claim 79 is Faulty

None of the three components identified in the Office Action meet the "linkage return address" recited in claim 79.

> Robinson '095 describes the "native dynamic link" or "dylnk" at 26:56-60:
>
> The fourth entry in the header 216a-216c is the native dynamic link 220d. The native dynamic link field is a pointer to the previous shadow frame header 214. Thus, in FIG. 22, the value stored in the field "dylnk" corresponds to the location of the next shadow frame header 216b.

The Office Action makes no attempt to show that the "dynamic link" has anything to do with an "address being deliberately chosen so that an attempt to execute ... will raise a program execution exception" and no pertinence is apparent.[9]

Robinson's "native routine return address" is chosen so that execution may resume there, without raising an exception. Robinson '095 26:53-55. Thus there is no "deliberate" exception.

The "non-native return address" is likewise chosen to be an address at which execution may legally resume in the software interpreter. Robinson '095 26:9-13. There is no teaching of a "deliberate" exception associated with the "non-native return address."

Thus, none of the three correspond to the "linkage return address ... deliberately chosen so that an attempt to execute ... will raise a program execution exception" of claim 79.

---

[9] The argument of Applicant's Response of 12/30/2005 regarding "dylnk" is withdrawn, to the extent that in some cases a pointer to a frame of data, which in turn contains an address deliberately chosen to raise an exception, might be equivalent to the claim.

### 3. Conclusion as to Claims 79-81

No rejection exists, and claim 79 distinguishes the art.

Claims 80-81 are dependent on claim 79, and are patentable with independent claim 79. In addition, these dependent claims recite additional features that further distinguish the art.

In the event that a future paper presents a statement of rejection, Applicant reserves the right to reply to the late examination, for example, by noting that finality must be withdrawn in light of the new ground of rejection, or by designating claims that stand or fall separately.

## E. Claims 5-13, 15-24, 26, 27, 30, 31, 82, 83

Claim 5 is discussed in connection with Robinson '095 and Bitar '460 at ¶ 31 of the Office Action. Claim 5 recites as follows:

> 5. A method, comprising:
>
> scheduling concurrent threads of control by a pre-existing thread scheduler of a computer, each thread having an associated context, an association between a thread and a set of computer resources of the associated context being maintained by the thread scheduler; and
>
> without modifying the thread scheduler, maintaining an association between one of the threads and an extended context of the thread through a context change induced by the thread scheduler, **the extended context including resources of the computer associated with the thread that are beyond those resources whose association with the thread is maintained by the thread scheduler**.

The Office Action is incomplete. For example, it misquotes the claim, and leaves out a portion that is bolded above – **even though that language has been at issue during the entire pendency of the application**. As best understood, perhaps the Office Action suggests that the bolded language of claim 5 might be met by unspecified "registers" of Robinson '095.

The first paragraph of claim 5 recites a "context" and a "pre-existing thread scheduler" that are designed to work with each other. The second paragraph of claim 5 then recites an "extended context" that is "beyond" the capabilities of the "thread scheduler." Claim 5 recites that, even though the "thread scheduler" is unable to manage the "extended context," nonetheless the scheduler and extended context are made to work with each other by "maintaining an association" between the "extended context" "through a context change induced by the thread scheduler." In some embodiments, claim 5 might allow the use of a mature operating system

(*e.g.*, Microsoft Windows) designed for one computer architecture to be used with a computer of a different architecture that has "extended" resources.

One example is shown in Figs. 3j, 3k, an 3m, at references 355, 360, and 370, and discussed in the specification at pages 42-44.

### 1. First Error: The Office Action's Analysis of the Prior Art Relative to Claim 5 is Faulty

There is no case in which Robinson has an "extended context including resources ... beyond those resources ... maintained by the thread scheduler" that maintains an "association" through a "context change induced by the thread scheduler." Rather, Robinson '095 has one set of X86 contexts for his X86 thread scheduler, and a non-overlapping set of Alpha contexts for his Alpha scheduler.. For example, the "context data structure" 180 used by Robinson's X86 thread scheduler (Robinson '095 25:10-60), on which the Office Action relies, is never mentioned anywhere else in Robinson's specification, let alone in combination with an Alpha thread scheduler.

The Office Action makes no attempt to compare Bitar '460 to this portion of claim 5.

As discussed next, the Office Action is simply silent on the relevant issue.

### 2. Second Error: The Office Action is Procedurally Insufficient to Raise any Rejection

The Office Action fails to comply with the procedural minima set out in 37 C.F.R. § 1.104(c)(2) and MPEP § 2141-2144.09 necessary to raise a rejection. Even though the same claim language has been at issue throughout prosecution, **the Office Action designates nothing in any reference to correspond to the bolded claim language**. Indeed, the Office Action explicitly ignores the claim language, by entirely omitting a portion of the claim language. Until the Office Action makes some minimal attempt to show that the bolded claim language is suggested in a reference, no *prima facie* rejection exists.

The Office Action designates a number of unrelated portions of Chernoff '028, without explaining the "pertinence" of the indicated portions. For example, col. 16, col. 25, and col. 27 are from the three different portions of Robinson '095, as discussed in §§ V.A.1, V.A.2 and V.A.3 above – the Action cites them with no attempt to interrelate the different components

discussed in the three sections. This fails to comply with 37 C.F.R. § 1.104(c)(2), which requires both the designation of the "particular portions relied on" and a clear explanation of "pertinence" of those portions.[10]

The Office Action makes no showing that distantly-separated portions of Chernoff '028 have any relationship to each other, and makes no showing of "motivation to combine" them. Nor does the Office Action make any showing of "reasonable expectation of success." Without the showings required by MPEP §§ 2143.01 and 2143.02, no *prima facie* rejection exists.[11]

The Office Action confuses the basic principles of obviousness: the different elements of obviousness are mixed together, and the order in which they come into play is disregarded. MPEP §§ 2142 and 2143 teach that obviousness has several separate sub-issues:

- "Analogous and non-analogous art," MPEP § 2141.01(a)
- "Suggestion or Motivation to Modify the References," MPEP § 2143.01
- "Reasonable expectation of success is required," MPEP § 2143.02
- "All claim limitations must be taught or suggested," MPEP § 2143.03
- Only after an examiner makes all three of the § 2143 *prima facie* showings does any burden shift to an applicant to make showings on the secondary factors. MPEP § 2142.

The Office Action, at ¶ 11, responds to an argument directed to directed to lack of showing of "reasonable expectation of success" as if the issue were the secondary factors of "superior or unexpected results" – but no *prima facie* showing was even made, and therefore no burden shifted. ¶ 11 responds to an argument of "reasonable expectation of success" as if it the issue were "non-analogous art" and "motivation to combine." The Office Action is irrelevant, and remains silent on any showing of "reasonable expectation of success" and "motivation to combine" the different portions and different elements of Robinson '095.

---

[10] A rejection may not combine different disclosures drawn from different parts of a reference without making a formal showing. *North American Oil Co. v. Star Brite Distributing, Inc.*, 46 Fed. Appx. 629, 631 (Fed. Cir. 2002) (district "court erred in considering that disparate disclosures in different parts of a multiple volume treatise ... can be combined to find anticipation"); *Ex parte Beuther*, 71 USPQ2d 1313, 1316 (Bd. Apt. App. & Interf. 2003) ("It is well-settled however, that anticipation is not established if ... it is necessary to pick, choose, and combine various portions of the disclosure not directly related to each other by the teachings of the reference").

[11] Paragraph 98 of the Office Action misparaphrases the argument and the MPEP test for obviousness. "Motivation to combine" and "reasonable expectation of success" are two distinct elements – one does not show one in order to establish the other. MPEP §§ 2143, 2143.01 and 2143.02.

The analysis in the Office Action is too incomplete and confused to constitute a rejection, and the rebuttal arguments miss the point.

At ¶ 9(1), the Office Action asserts that claim language set forth in bold above is "not recited in ... the claim." The Office Action is wrong. The source of the Examiner's view may be that ¶ 31(3) of the Office Action misquotes the claim, by omitting the last six words.

At ¶ 9(2), the Office Action asserts that Bitar '416 meets some part of the claim, though not the "extended context" portion that was argued. The April 2006 Office Action fails to answer the specific material traversed.

The attempts to "answer all material traversed" are either wrong or irrelevant.

### 3. Claims 6-13, 15-24, 26, 27, 30-31, 82 and 83 are Patentable with Claim 5

No rejection exists, and claim 5 distinguishes the art.

Claims 6-13, 15-24, 26, 27, 30-31, 82 and 83 are dependent on claim 5, and are patentable with independent claim 5. In addition, these dependent claims recite additional features that further distinguish the art.

In the event that a future paper presents a statement of rejection, Applicant reserves the right to reply to the late examination, for example, by noting that finality must be withdrawn in light of the new ground of rejection, or by designating claims that stand or fall separately.

## F. Claims 56-75

Claim 56 is mentioned in ¶ 63 of the Office Action, in the context of Robinson '095 combined with Fleck '641. Claim 56 recites as follows:

> 56. A method, comprising:
>
> without modifying a pre-existing operating system of the computer, establishing an entry handler for execution at a specified entry point or on a specified entry condition to the operating system, the entry handler programmed to save a context of an interrupted thread and modify the thread context before delivering the modified thread context to the operating system;
>
> without modifying the operating system, establishing an exit handler for execution on resumption from the operating system following an entry through the entry handler, the exit handler programmed to restore the context saved by a corresponding execution of the entry handler.

At least on the current state of the record, the same portions of the specification discussed in connection with claim 33, § VI.C, above, are relevant to claim 56.

### 1. The Office Action Is Too Imprecise and Contains too Many Errors To Raise Any Rejection or to Permit a Response

The Office Action is vague, mixes and matches unrelated parts of the Robinson '095 reference, offers a number of misquotations, and omits consideration of elements of *prima facie* obviousness.

The Office Action never identifies any particular component that corresponds to the "entry handler" throughout the claim. Similarly, the Office Action takes no consistent position on the "exit handler." Rather, the Office Action matches up one thing here, another there.

Throughout ¶ 63, the Office Action refers to "the exception handler" of Robinson '095. There are several "exception handlers" mentioned in Robinson '095, but the Office Action identifies none of them. Is the Examiner mixing and matching them, or is one in particular thought to be pertinent? Which one?

At ¶ 63(2), the Office Action asserts that Robinson '095 teaches that "background system 34 or the interpreter 44, or part of [an unidentified] exception handler ... translate from non-native to native or temporary storage for logical register manipulations, e.g." This misrepresents the content of Robinson '095: the Office Action has glued together two bits of two unrelated sentences. One bit comes from Robinson '095 11:10-20, which reads as follows:

> The background system 34 will, in general, translate nearly all instructions provided from the non-native applications stored in 17b...

and the other bit comes from sixteen columns away, Robinson '095 27:1-15, which reads as follows:

> Accordingly, the compiler often provides temporary storage for logical register manipulations and uses the program stack to store these registers.

Note that it is the "compiler" acting here, not the "background system 34 or the interpreter 44, or part of the exception handler" as stated in the Office Action. Gluing together unrelated bits and pieces of a reference is impermissible. The law requires that references are to be read "fairly." *In re Schaub*, 1991 WL 252968 at **2, 1991 U.S. App. Lexis 28164 at *5 (Fed. Cir. Nov. 27,

1991) (non-precedential) (conclusions of anticipation can only be based on "a fair reading of the reference as a whole").[12]

The Office Action also fails to recognize the fundamental divide among the different sections of Robinson '095: Robinson's Alpha environment handles only Alpha context, and Robinson's X86 environment only handles X86 context. See § V.A, above. Paragraph 63(2), discussing the first paragraph of claim 56, compares the "entry to the operating system" of claim 56 to "entry to second architecture from first architecture, etc." Then, the language "save a context" of claim 56 is compared to "current state saved in context data structure" of Robinson '095. These two statements are incompatible. Robinson '095 25:10-60 teaches that the "context data structure" 180 is only used to switch among X86 threads, never "to [switch to] second architecture from first architecture." In contrast, the portions of Robinson '095 directed to "entry to second architecture from first architecture," largely Robinson '095 25:62 to 36:36, never mention context data structure 180. The analysis of Robinson '095 in the Office Action is simply wrong.

The Office Action is not clear. It several times uses "etc." and "e.g." What does this mean – is there something else in the reference that is relevant, or not? If so, what?

The Office Action fails to make appropriate showings of "motivation to combine" or "reasonable expectation of success." How could one apply the "hardware assisted" techniques of Fleck '641 (Fleck '641 1:37) to the pure software system discussed in the cited portions of Robinson '095 to achieve any meaningful result whatsoever? What would be the basis for "reasonable expectation of success?" How would one be motivated to back-graft new hardware onto old software, without rendering the prior art unsuitable for its intended purpose? Without the showings required by MPEP §§ 2143.01 and 2143.02, no *prima facie* rejection exists.

The Office Action does not suggest that Bitar '460 overcomes any of the above defects in the consideration of Robinson '095 and claim 56.

The Office Action is too procedurally incomplete and technologically garbled to raise a rejection, or to permit any direct response.

---

[12] See footnote 10.

### 2. Conclusion as to Claims 56-75

No rejection exists, and claim 56 distinguishes the art.

Claims 57-75 are dependent on claim 56, and are patentable with independent claim 56. In addition, these dependent claims recite additional features that further distinguish the art.

In the event that a future paper presents a statement of rejection, Applicant reserves the right to reply to the late examination, for example, by noting that finality must be withdrawn in light of the new ground of rejection, or by designating claims that stand or fall separately.

## VII. Conclusion

In view of these remarks, Applicant respectfully submits that the claims are in condition for allowance. Applicant requests that the application be passed to issue in due course. The Examiner is urged to telephone Applicant's undersigned counsel at the number noted below if it will advance the prosecution of this application, or with any suggestion to resolve any condition that would impede allowance. In the event that any extension of time is required, Applicant petitions for that extension of time required to make this response timely. Kindly charge any additional fee, or credit any surplus, to Deposit Account No. 23-2405, Order No. 114596-05-4013.

Respectfully submitted,

WILLKIE FARR & GALLAGHER LLP

Dated: June 20, 2006

By: _____

David E. Boundy
Registration No. 36,461

WILLKIE FARR & GALLAGHER LLP
787 Seventh Ave.
New York, New York 10019
(212) 728-8000
(212) 728-8111 Fax